REMARKS

This Amendment is filed in response to the FINAL Office Action mailed December 10, 2008. A Request for Continued Examination and the associated fee is also filed herewith. All objections and rejections are respectfully traversed.

Claims 1-75 are in the case.

Claims 73-75 have been added.

Claims 1, 16, 28, 34, 39, 41, 44, 49, 54, 69, 71 have been amended.

Interview Summary

Applicant would like to thank Examiner Portka for conducting the Applicant Initiated Interview on March 9, 2009 and for helping to advance this Application closer to allowance. It should be noted that the interview focused on a proposed new claim (i.e., 73). The rejected claims have been amended to include similar limitations. For simplicity reasons, this Amendment is argued in terms of the new claim discussed during the interview. Generally, the issue discussed was the use of the term "read stream" as illustratively used in Applicant's claimed invention and mentioned in Applicant's background.

Applicant's exemplary use of the term is cited, in part, below:

> A *read stream* is defined as a set of one or more client requests that instructs the storage system to retrieve data from a logically contiguous range of file offsets within a requested file. In other words, after the read stream's first request is received, every subsequent client request in the read stream logically "extends" a contiguous sequence of file offsets in the file accessed by the stream's previous request. Accordingly, a read stream may be construed by the file system as a sequence of client requests that directs the storage system to retrieve a sequence of data blocks assigned to consecutively numbered fbns. (Specification, p. 4, lines 1-8)

Applicant has incorporated similar language in the claims to help clarify the term's usage. Examiner stated that this clarification appeared to overcome the cited prior art reference, but that a closer look at the reference, as well as a new search, would be required.

Rejection Under 35 U.S.C. §112

At paragraph 4 of the Office Action, claims 16-21 were rejected under 35 U.S.C §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter to which applicant regards as the invention. Claim 16 has been amended and is believed to be in condition for allowance. Accordingly, claims 16-21 are believed to be in condition for allowance.

Rejection Under 35 U.S.C. §102(b)

At paragraph 6 of the Office Action, claims 1-14 and 16-72 were rejected under 35 U.S.C. §102(b) as being unpatentable in view of Permut U.S. Patent No. 6,260,115 issued on July 10, 2001 (hereinafter "Permut").

Applicant's claimed invention, as set forth in representative *new* claim 73, comprises in part:

> 73. A method for operating a computer data storage system, comprising:
>
> **receiving a first data read command *associated with a particular read stream***, wherein the read stream is a set of one or more client read commands to retrieve data from a contiguous range of file offsets within a requested file;
>
> determining one or more input parameters of the first data read command;
>
> establishing, in response to the input parameters, a readahead hint, wherein the readahead hint determines a number of data blocks to readahead;
>
> **receiving a next data read command *associated with the particular read stream***;
>
> determining one or more input parameters of the next data read command;
>
> ***modifying*, in response to the next input parameters, *the readahead hint* to obtain a *modified* readahead hint**; and
>
> ***adjusting*, *in response to the modified readahead hint*, the number of data blocks to readahead for the next read command *associated with the particular read stream***, wherein the adjusted number of data blocks is stored in memory.

Permut discloses using factors to determine a number of tracks to prestage for multiple read requests associated with a single data stream (col. 3, lines 66 to col. 4, line 3). While different factors may vary the prestage number from one data stream to another data stream, Permut shows that read requests associated with any *one read stream* will have the *same* prestage amount (col. 10, lines 47-52). Once the factors determine the number of tracks to prestage for each client request associated with any one particular read stream, that number is *remembered* (i.e., *fixed*) in a temporary storage location to be applied to subsequent read requests associated with the particular read stream (col. 10, lines 47-52).

Applicant respectfully urges that Permut does not disclose Applicant's claimed novel **receiving a first data read command *associated with a particular read stream*, receiving a next data read command *associated with the particular read stream*, *modifying*, in response to the next input parameters, *the readahead hint* to obtain a *modified* readahead hint**, and ***adjusting*, *in response to the modified readahead hint*, the number of data blocks to readahead for the next read command *associated with the particular read stream*.**

Applicant claims **receiving a first data read command** ***associated with a particular*** (e.g., same) ***read stream***, wherein the **read stream** is a *set* (e.g., group) of one or more client read commands to retrieve data from a contiguous range of file offsets within a requested file. As discussed in the interview, typical readahead algorithms may use factors or hints which determine or adjust the readahead amounts for read commands associated with a plurality of read streams, however, those algorithms maintain the *same* hint which determines the readahead amounts for <u>each</u> read command comprising a particular read stream (e.g., a particular set of client read commands associated with the *same* read stream). However, Applicant claims determining a readahead amount (e.g., **readahead hint**) for a first read command ***associated with*** e.g., a ***particular read stream***, and **receiving a next data read command *associated with the*** (same) ***particular read stream***, where **input parameters of** e.g., **the next read command** may be used to ***modify the readahead hint*** **to obtain a *modified* readahead hint**. By ***modifying*** e.g., ***the readahead hint*** associated with ***the*** (same) ***particular read stream***, it is possible to

adjust the data blocks to readahead for read commands, even when the read commands are ***associated with the*** (same) ***particular read stream***.

As mentioned above, Permut uses factors to determine and assign a *fixed* prestage amount for data requests associated with a particular read stream and further shows using different factors to adjust and assign a different *fixed* prestage amount to data requests associated with a different read stream. However, using factors to adjust one *fixed* prestage amount (for all read requests associated with one read stream) to another *fixed* prestage amount (for all read requests associated with another read stream) is not the same as Applicant's claimed **adjusting the number of data blocks to readahead for the next read command *associated with the*** (same) ***particular read stream***. Specifically, Permut does not adjust the number of tracks to prestage for individual read commands if those individual read commands are associated with the <u>same</u> read stream. In other words, Permut does not show **adjusting the number of data blocks to readahead for** *individual* **data read commands associated with a particular read stream** once the factors determine and assign the prestage amount for that ***particular read stream***. In contrast, Applicant claims receiving multiple **data read commands *associated with*** the ***same particular read stream***, where each data read command (i.e., **first and next**) ***associated with the*** *same* ***particular read stream*** may have an adjusted (i.e., different) **number of readahead data blocks**. As such, Permut is totally silent to Applicant's claimed **adjusting the number of data blocks to readahead for the next read command *associated with the*** (same) ***particular read stream*** because Permut only shows varying the *fixed* prestage amount between individual read streams.

However, even if Permut does show adjusting the prestage amount for the same read stream, Permut is still totally silent to Applicant's claimed ***modifying*, in response to the next input parameters, *the readahead hint*** (e.g., readahead metadata) **to obtain a *modified* readahead hint**. In particular, Permut shows adjusting the prestage amount simply by using different factors in the system to create a *new* hint. In total contrast, Applicant claims ***modifying the*** (already existing) ***readahead hint*** (e.g., ***in response to the input parameters of the next read command***) **to obtain a *modified* readahead hint** that is used to determine the number of readahead blocks for the next read command. As such, even if Permut does show adjusting the prestage amount for the same read stream, Permut is still totally silent to adjusting the prestage amount for the same read stream by ***modifying the*** (already existing) ***readahead hint*** **to obtain a *modified* readahead hint**, because Permut creates an entirely new hint.

Accordingly, Applicant respectfully urges that the Permut patent is legally precluded from anticipating the claimed invention under 35 U.S.C. § 102 because of the absence from the Permut patent of Applicant's claimed novel **receiving a first data read command *associated with a particular read stream*, receiving a next data read command *associated with the particular read stream*, *modifying*, in response to the next input parameters, *the readahead hint* to obtain a *modified* readahead hint**, and ***adjusting, in response to the modified readahead hint,* the number of data blocks to readahead for the next read command *associated with the particular read stream*.**

Applicant's Interpretation of the Prior Art

Applicant's interpretation of Permut was derived, in part, from the following excerpts:

> In the present invention, the desired number of tracks to prestage ahead is *remembered in a temporary storage location*. This value is typically the greater number of the additional tracks indicated with a domain prestaging hint, or a *fixed distance* to prestage ahead for other types of sequential hints. In the preferred embodiment, a host sequential hint would ordinarily result in staying *6 tracks ahead of the current access*, unless the hint is for a special case such as but not limited to a partitioned data set search assist command sequence or for a special high speed data mover command chain, in which cases the control unit would prestage 15 tracks ahead of the current access. (col. 10, lines 47-52) (emphasis added)

> When deciding the number of tracks to prestage ahead of the current access, defined as the prestage factor, prestaging processes may take into account host prestaging or sequential hints, detection of very long sequential patterns or even the amount of cache memory configured in the subsystem. The information *contained in the list entries* may be used to demote from cache those tracks that lag a certain address range behind the current address in a sequential stream. (col. 3, line 66 – col. 4, line 6) (emphasis added)

> Effective prestaging methods must coordinate the use of sequential hints *presented directly by the host processors*, implied hints *contained in the channel programs*, and sequential detection methods where *no hints* are present. Effective

> prestaging methods must also record which tracks have *already been prestaged* to avoid requesting the prestaging of the same tracks multiple times which would waste valuable storage subsystem processing cycles and possibly even back-end bandwidth. (col. 2, lines 18-26) (emphasis added)

Rejection under §103(a)

At paragraph 30 of the Office Action, claim 15 was rejected under 35 U.S.C. §103(a) as being unpatentable over Permut in view of Vishlitzky et al., U.S. Patent No. 5,649,156, issued on July 15, 1997 (hereinafter "Vishlitzky").

Claim 15 is dependent from independent claim 1 which is believed to be in condition for allowance. As such, claim 15 is believed to be in condition for allowance.

Claim Support

All new claim proposals as well as any proposed claim amendments are believed to be fully supported by Applicant's specification. Upon request, additional citations may be provided.

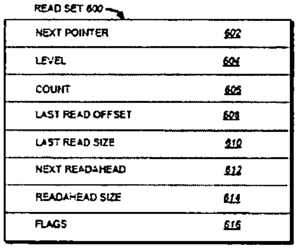
With reference to the claim(s), support for the proposed new and amended claims may be found, in part, at Specification page 25, line 9-19:

> Fig. 8 is a schematic block diagram of an illustrative table 800 that may be used by the file system 260 to adaptively select a readahead size 614 associated with a read stream. Specifically, in response to receiving a client read request *in the read stream*, the file system can select a number of readahead data blocks 820 to speculatively retrieve based on the amount of client-requested data 810 requested in the received request. By way of example, *if* the client requests less than 64 kB of data, *then* the file system sets the readahead size 614 equal to two times a predetermined number N of data blocks, e.g., where N equals 32 data blocks. *If* the client requests more than 64 kB and less *than* 128 kB, then the readahead size 614 is set equal to four times the predetermined number of data blocks. Similarly, *if* the client requests between 128 kB and 256 kB of data, *then* the readahead size 614 is set equal to six times the predetermined number of data blocks. (emphasis added)

With reference to the claim(s), support for the proposed new and amended claims may be found, in part, at Specification page 6, line 18 – page 7, line 2:

According to an illustrative embodiment, the file system manages a *separate set of readahead metadata for each read stream.* Consequently, the file system can *individually adjust readahead operations for each read stream* by appropriately modifying configuration parameters *stored in the read stream's associated set of metadata.* For instance, each read stream's associated set of readahead metadata may include, *inter alia*, an indication of when readahead operations are performed, an amount of readahead data to retrieve and an indication of how long the retrieved readahead data should be retained in buffer cache. Advantageously, the contents of the read stream's associated set of readahead metadata may be dynamically adjusted as client requests are processed so as to optimize the amount of readahead data that is prefetched and minimize cache pollution in the storage system… Further to the illustrative embodiment, *each set* of readahead metadata is stored in a corresponding *readset* data structure. (emphasis added)

With reference to the claim(s), support for the proposed new and amended claims may be found, in part, by Figs. 6, 7, and 8:



NEXT POINTER	602
LEVEL	604
COUNT	606
LAST READ OFFSET	608
LAST READ SIZE	610
NEXT READAHEAD	612
READAHEAD SIZE	614
FLAGS	616

FIG. 6

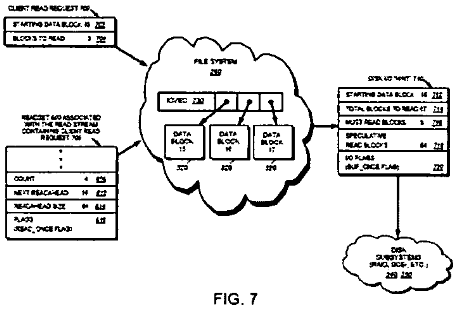
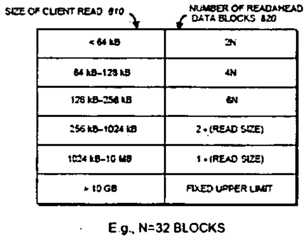

CLIENT READ REQUEST 700
STARTING DATA BLOCK 16 702
BLOCKS TO READ 3 704
READSET 600 ASSOCIATED WITH THE READ STREAM CONTAINING CLIENT READ REQUEST 700
COUNT 4 606
NEXT READAHEAD 19 612
READAHEAD SIZE 64 614
FLAGS (READ_ONCE FLAG) 616
FILE SYSTEM 260
IOVEC 730
DATA BLOCK 15
DATA BLOCK 16
DATA BLOCK 17
320
320
320
DISK I/O "HINT" 710
STARTING DATA BLOCK 16 712
TOTAL BLOCKS TO READ 67 714
MUST READ BLOCKS 3 716
SPECULATIVE READ BLOCKS 64 718
I/O FLAGS (BUF_ONCE FLAG) 720
DISK SUBSYSTEMS (RAID, DISK, ETC.) 240 250

FIG. 7


SIZE OF CLIENT READ 810
NUMBER OF READAHEAD DATA BLOCKS 820

< 64 kB	2N
64 kB–128 kB	4N
128 kB–256 kB	6N
256 kB–1024 kB	2 • (READ SIZE)
1024 kB–10 MB	1 • (READ SIZE)
> 10 GB	FIXED UPPER LIMIT

E.g., N=32 BLOCKS

FIG. 8

Conclusion

All newly proposed claims and proposed claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Respectfully submitted,

/Michael T. Abramson/

Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927